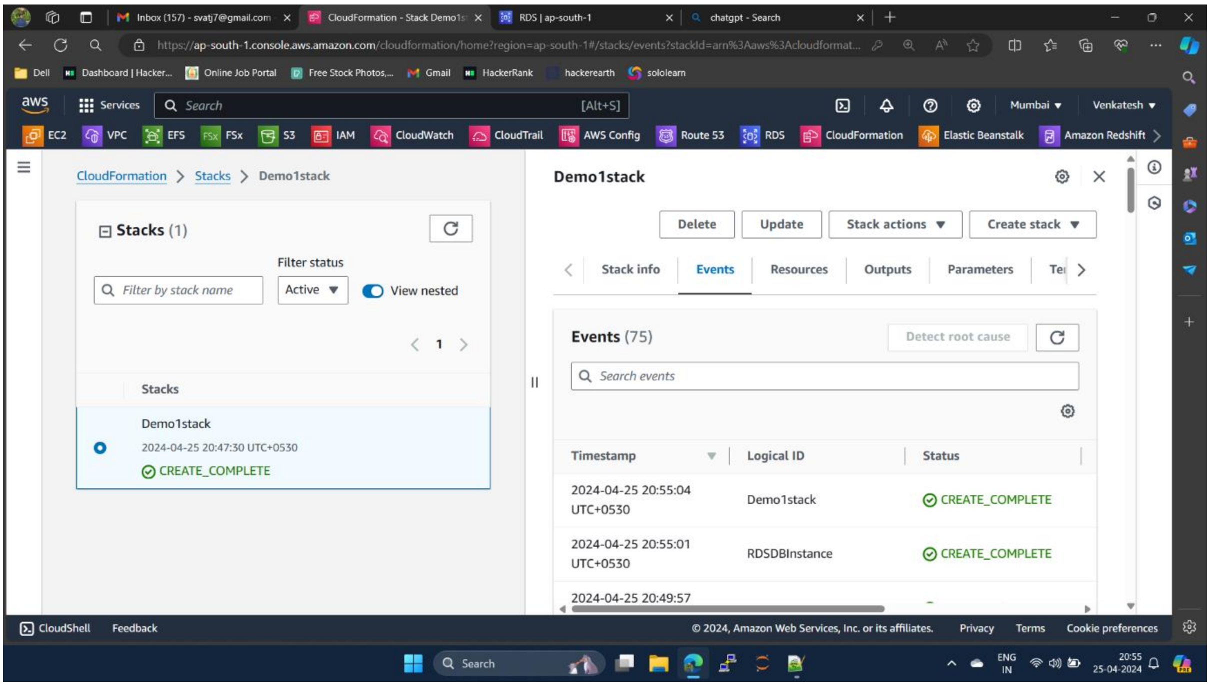
Multi-Tier-Architecture



https://ap-south-1.console.aws.amazon.com/cloudformation/home?region=ap-south-1#/stacks/events?stackId=arn%3Aaws%3Acloudformat...
CloudFormation > Stacks > Demo1stack
Delete
Update
Stack actions
Create stack
Stack info
Events
Resources
Outputs
Parameters
Detect root cause

Stacks (1)

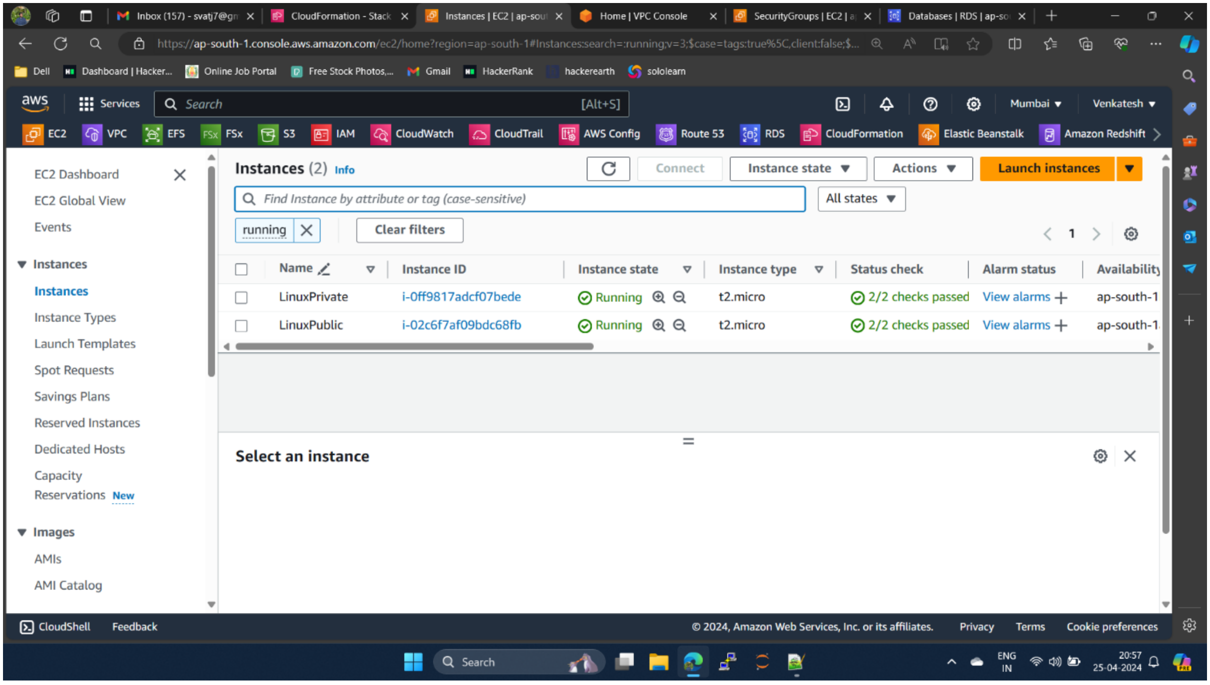
Filter status: Active — View nested

Stacks
● Demo1stack 2024-04-25 20:47:30 UTC+0530 CREATE_COMPLETE

Demo1stack

Events (75)

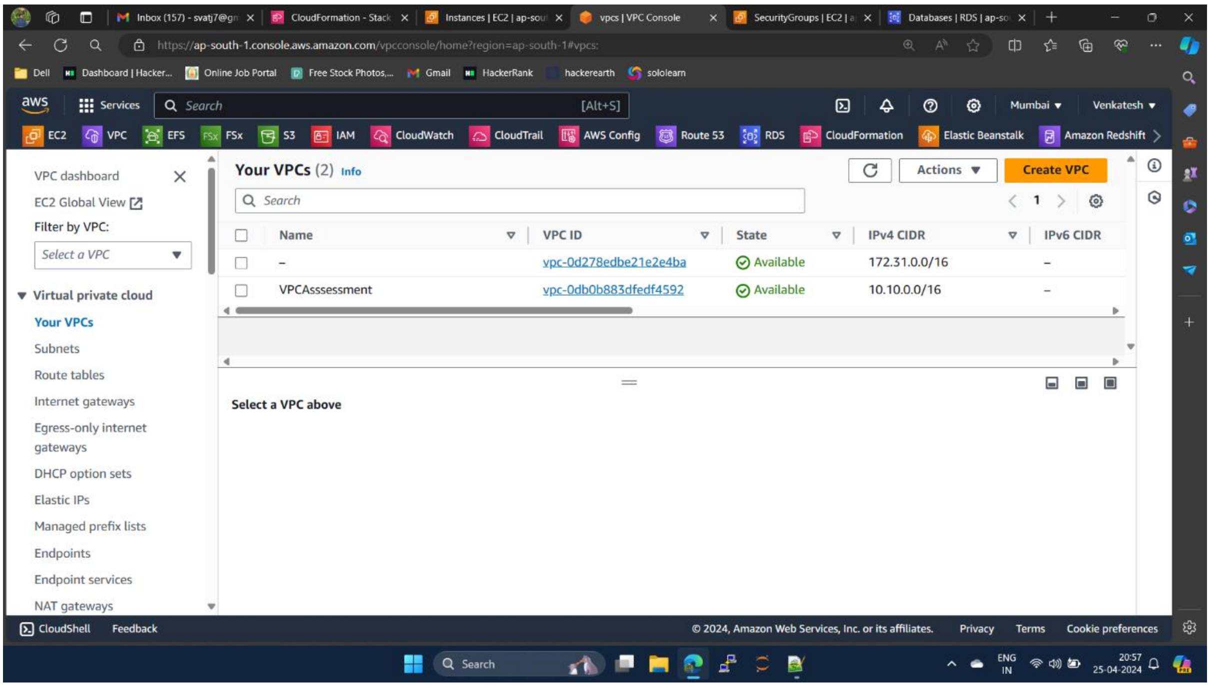
Timestamp	Logical ID	Status
2024-04-25 20:55:04 UTC+0530	Demo1stack	CREATE_COMPLETE
2024-04-25 20:55:01 UTC+0530	RDSDBInstance	CREATE_COMPLETE
2024-04-25 20:49:57		



Instances (2) Info

Filter: running

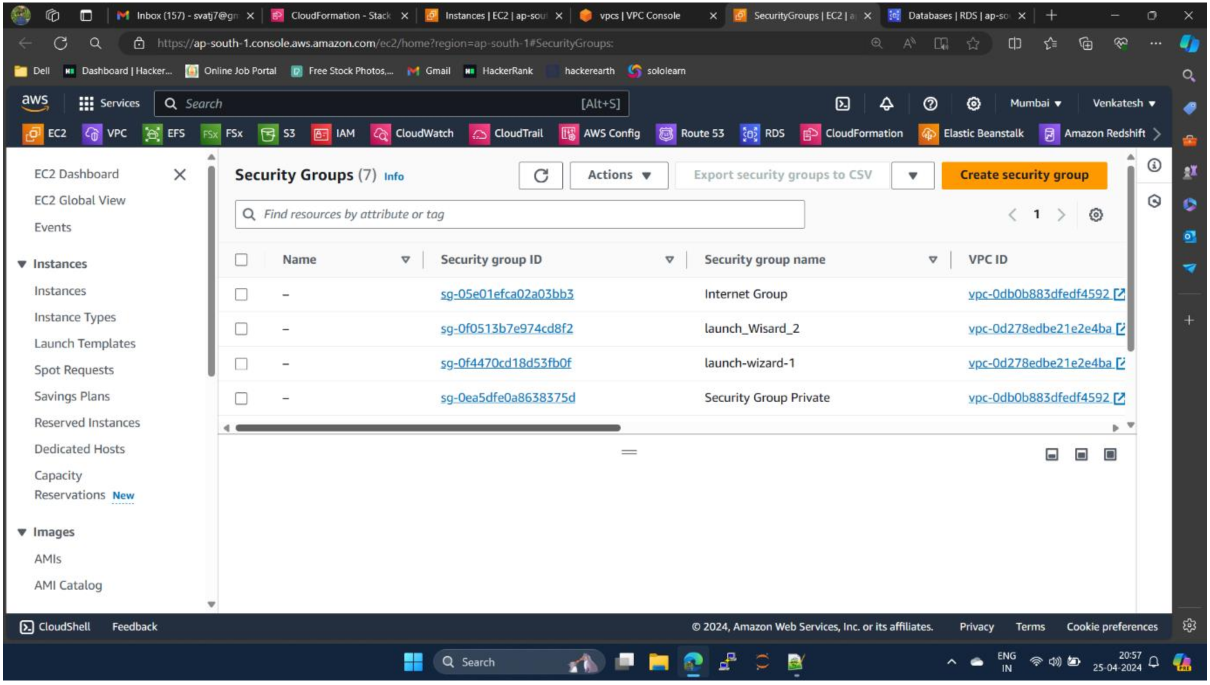
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
LinuxPrivate	i-0ff9817adcf07bede	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1
LinuxPublic	i-02c6f7af09bdc68fb	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1



Your VPCs (2) Info

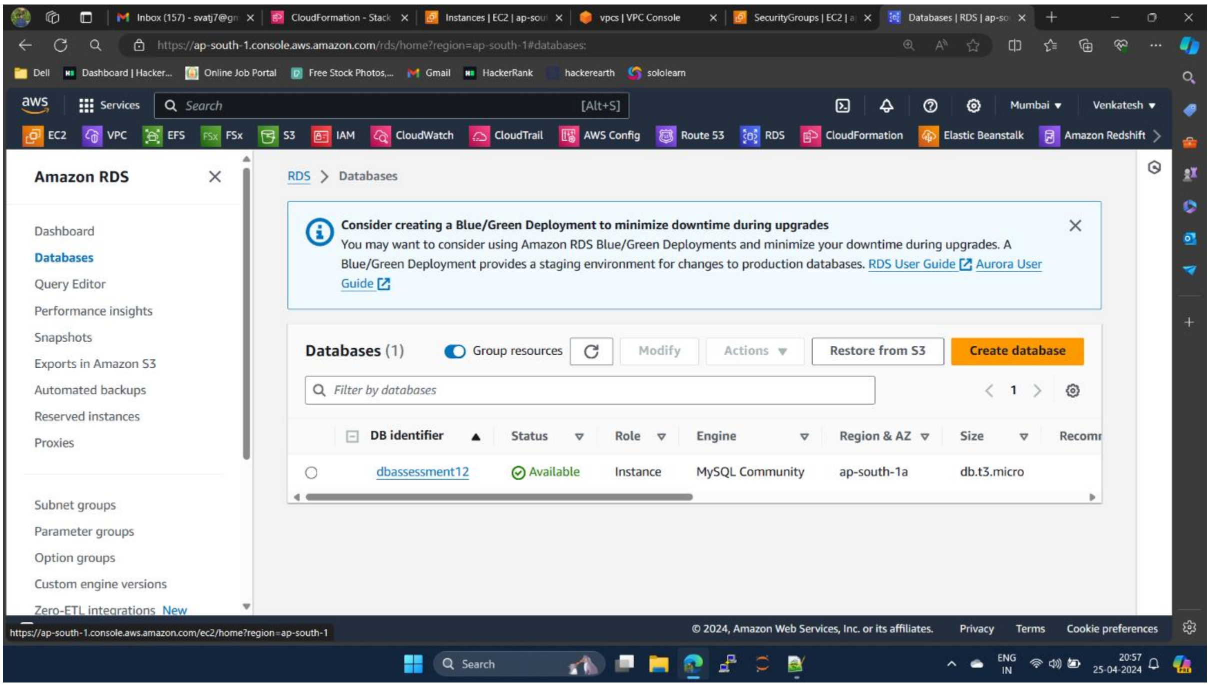
□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
□	–	vpc-0d278edbe21e2e4ba	✓ Available	172.31.0.0/16	–
□	VPCAsssessment	vpc-0db0b883dfedf4592	✓ Available	10.10.0.0/16	–

Select a VPC above



Security Groups (7) Info

□	Name	Security group ID	Security group name	VPC ID
□	–	sg-05e01efca02a03bb3	Internet Group	vpc-0db0b883dfedf4592
□	–	sg-0f0513b7e974cd8f2	launch_Wisard_2	vpc-0d278edbe21e2e4ba
□	–	sg-0f4470cd18d53fb0f	launch-wizard-1	vpc-0d278edbe21e2e4ba
□	–	sg-0ea5dfe0a8638375d	Security Group Private	vpc-0db0b883dfedf4592

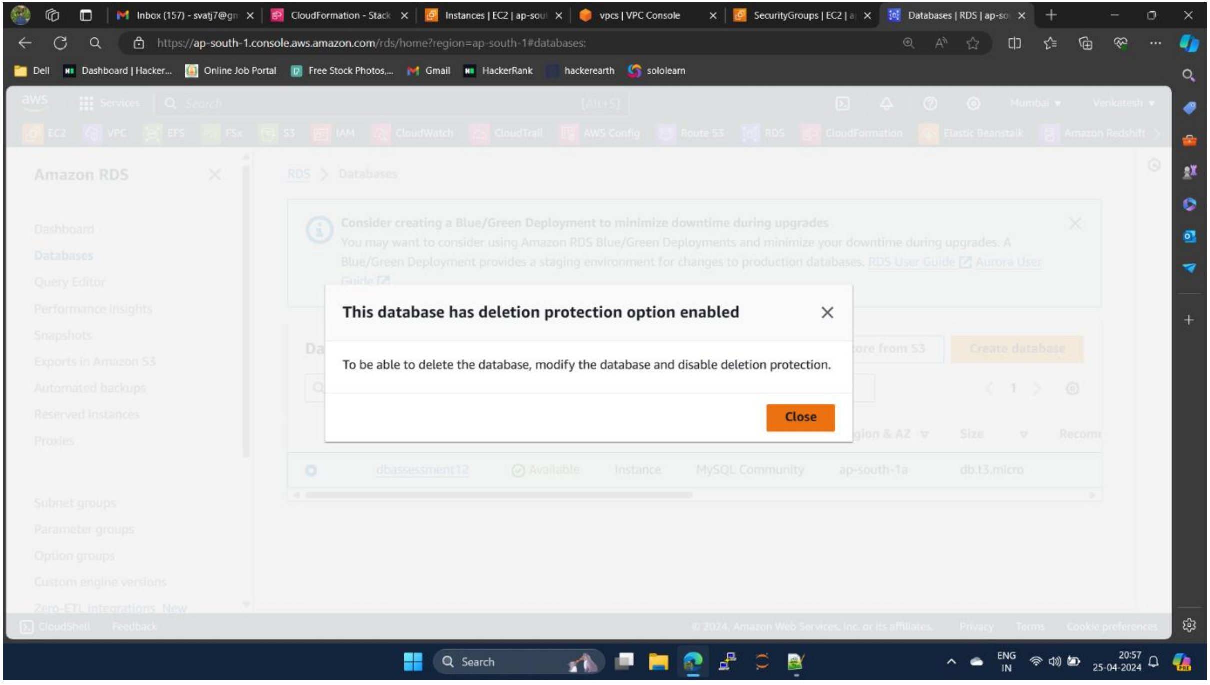


Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide

Databases (1)

	DB identifier	Status	Role	Engine	Region & AZ	Size	Recomm
○	dbassessment12	✓ Available	Instance	MySQL Community	ap-south-1a	db.t3.micro	

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



This database has deletion protection option enabled

To be able to delete the database, modify the database and disable deletion protection.

YAML Code:

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  InstanceTypeParameter:
    Type: String
    Default: t2.micro
    Description: Enter instance size. Default is t2.micro.
  AMI:
    Type: String
    Default: ami-001843b876406202a
    Description: The Linux AMI to use.
  Key:
    Type: AWS::EC2::KeyPair::KeyName
    Description: Select from Existing Keys.

  MasterUsername:
    Type: String
    Description: The username for the database.

  MasterUserPassword:
    Type: String
    Description: The password for the database.
    "NoEcho": true

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.10.0.0/16
      EnableDnsSupport: true
    EnableDnsHostnames: true
    InstanceTenancy: default
    Tags:
      - Key: Name
        Value: VPCAsssessment
InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Name
        Value: InternetGatewayAssessment
VPCGatewayAttachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    VpcId: !Ref VPC
    InternetGatewayId: !Ref InternetGateway

#Public Subnet
SubnetA:
  Type: AWS::EC2::Subnet
  Properties:
    AvailabilityZone: !Select [ 0, !GetAZs ]
    VpcId: !Ref VPC
    CidrBlock: 10.10.1.0/24
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: PublicSubnetAssessment
PublicRouteTable:
  Type: AWS::EC2::RouteTable
```

```yaml
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: RouteTablePublicSubnet
PublicInternetRoute:
  Type: AWS::EC2::Route
  DependsOn: VPCGatewayAttachment
  Properties:
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
    RouteTableId: !Ref PublicRouteTable
SubnetARouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref SubnetA

#Private Subnet
SubnetB:
  Type: AWS::EC2::Subnet
  Properties:
    AvailabilityZone:  !Select [ 1, !GetAZs ]
    VpcId: !Ref VPC
    CidrBlock: 10.10.2.0/24
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: PrivateSubnetAssessment
```

```
# A NAT Gateway:
NATGateway:
 Type: AWS::EC2::NatGateway
 Properties:
   AllocationId: !GetAtt ElasticIPAddress.AllocationId
   SubnetId: !Ref SubnetA
   Tags:
   - Key: Name
     Value: NatGetwayAssessment
ElasticIPAddress:
  Type: AWS::EC2::EIP
  Properties:
    Domain: VPC

RouteTablePrivate:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: RouteTablePrivateSubnet
NATRoute:
  DependsOn: NATGateway
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref RouteTablePrivate
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NATGateway

SubnetBRouteTableAssociationPrivate:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTablePrivate
    SubnetId: !Ref SubnetB

InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "Internet Group"
    GroupDescription: "SSH and web traffic in, all traffic out."
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
        CidrIp:  0.0.0.0/0
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        CidrIp:  0.0.0.0/0
    SecurityGroupEgress:
      - IpProtocol: -1
        CidrIp: 0.0.0.0/0

InstanceSecurityGroupPrivate:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "Security Group Private"
    GroupDescription: "SSH from the Public Subnet"
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
        CidrIp:  10.10.1.0/24
    SecurityGroupEgress:
      - IpProtocol: -1
        CidrIp: 0.0.0.0/0

InstanceSecurityGroupDataBase:
  Type: "AWS::EC2::SecurityGroup"
  Properties:
    GroupDescription: "Database instances security group"
    VpcId: !Ref VPC
    SecurityGroupIngress:
        - IpProtocol: tcp
          CidrIp: 10.10.2.0/24
          FromPort: 3306
          ToPort: 3306
    SecurityGroupEgress:
        - IpProtocol: -1
          CidrIp: 0.0.0.0/0

RDSDBSubnetGroup:
  Type: "AWS::RDS::DBSubnetGroup"
  Properties:
      DBSubnetGroupDescription: "Subnet Group for mySQL database"
      DBSubnetGroupName: !Sub "${AWS::Region}-aws-database-subnet-group14"
      SubnetIds:
        - !Ref SubnetA
```

```
     - !Ref SubnetB
    Tags:
     - Key: Name
      Value: DBSubnetGroup

 RDSDBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      DBInstanceIdentifier: DBAssessment12
      AllocatedStorage: 20
      DBInstanceClass: db.t3.micro
      Engine: "MYSQL"
      MasterUsername: !Ref MasterUsername
      MasterUserPassword: !Ref MasterUserPassword
      MultiAZ: false
      EngineVersion: 8.0.35
      AutoMinorVersionUpgrade: true
      PubliclyAccessible: false
      StorageType: gp2
      Port: 3306
      StorageEncrypted: false
      CopyTagsToSnapshot: true
      EnableIAMDatabaseAuthentication: false
      DeletionProtection: true
      DBSubnetGroupName: !Ref RDSDBSubnetGroup
      VPCSecurityGroups:
       - !Ref InstanceSecurityGroupDataBase
      MaxAllocatedStorage: 1000
      Tags:
       - Key: Name
```

```
              Value: DBAssessment
            - Key: createdBy
              Value: Igor Silva
            - Key: Project
              Value: AssessmentModule7
            - Key: Environment
              Value: Prod

LinuxPublic:
  Type: 'AWS::EC2::Instance'
  Properties:
    SubnetId: !Ref SubnetA
    ImageId: !Ref AMI
    InstanceType: !Ref InstanceTypeParameter
    KeyName: !Ref Key
    SecurityGroupIds:
      - Ref: InstanceSecurityGroup
    Tags:
      - Key: Name
        Value: LinuxPublic

LinuxPrivate:
  Type: 'AWS::EC2::Instance'
  Properties:
    SubnetId: !Ref SubnetB
    ImageId: !Ref AMI
    InstanceType: !Ref InstanceTypeParameter
    KeyName: !Ref Key
    SecurityGroupIds:
      - Ref: InstanceSecurityGroupPrivate
      Tags:
      - Key: Name
        Value: LinuxPrivate

  HostedZone:
    Type: AWS::Route53::HostedZone
    Properties:
      HostedZoneConfig:
        Comment: "
      Name: newpracticedomain.ml

  MyDNSRecord:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref HostedZone
      Name: www.newpracticedomain.ml.
      Type: A
      TTL: 300
      ResourceRecords:
      - !GetAtt LinuxPublic.PublicIp

Outputs:
  PublicIp:
   Description: Server's PublicIp Address
   Value:
     Fn::GetAtt:
       - LinuxPublic
       - PublicIp
  HostedZoneID:
    Description: The ID of the Hosted Zone.
```

```
Value:
  Ref: HostedZone
```